TRECS (TRANSPONDER RECONFIGURATION SYSTEM)

Dilene Cruickshank
Integral Systems, Inc.
Lanham, MD 20706
dnelson@integ.com

Overview

TRECS is a sophisticated, labor-saving tool designed to support payload reconfiguration and command procedure generation in the satellite payload operations factory testing and in-orbit test environments. TRECS enables payload and test engineers to visualize and modify complex satellite system configurations. When the desired reconfiguration is set, TRECS generates a valid spacecraft procedure or command list that can be executed readily on the real-time ground system, such as ISI's (Integral Systems, Incorporated's) EPOCH T&C (Telemetry and Command), to implement the correct modifications required on the satellite. This method greatly reduces the labor required to develop payload procedures and can help to reduce payload outage durations significantly.

TRECS' user-configurable, model-driven design allows it to be used not just for one satellite, but for an entire satellite fleet, eliminating the need for costly mission-specific software changes. The TMD (TRECS Model Definer) is an independent, flexible model editing and construction tool enabling the assembled models to support multiple payload types. The TMD supports a wide variety of payload configuration objects, such as various types of switches, channel groups, and redundancy rings. Components in the model also can be flagged as failed or as reserved to guard against changes.

TRECS also can help users to reduce potentially costly communication outages significantly by minimizing the time required to perform nominal and recovery payload operations. TRECS enables users to graphically visualize and perform "what if" analysis on the payload configuration, its components, and all associated input/output paths. When a component is reconfigured, the user can see immediately how the result affects the signal paths throughout the entire diagram. The time that payload engineers spend planning reconfigurations is reduced significantly, thereby increasing efficiency and reducing operational workload.

To further reduce the time required to make large-scale changes or to help the user determine the best way to achieve a particular result, TRECS also offers an AutoSolve option for analysis, which can plan an optimal reconfiguration for payload subsets based on user-selectable rules and an algorithm. The AutoSolve option can plan a reconfiguration to limit or minimize the number of changes necessary on the satellite.

TRECS was designed with payload engineers in mind. For example, TRECS eliminates the effort required to develop separate, detailed executable procedures manually based on the planned reconfiguration results. TRECS generates an automatically formatted command procedure or a simple command list, both of which are ready instantly to be loaded and executed via the real-time satellite control system. All component changes are mapped directly to the commands and corresponding telemetry checks for the particular payload via the model definition.

TRECS helps users to allocate their time effectively and ensures immediate access to the information that they need. Many satellite payloads are highly complex, and defining payload configuration can easily become a time consuming task…especially following an unexpected, large-scale change, such as a load shedding anomaly. TRECS offers simple and effective tools to assist users in performing this task quickly and easily. When a telemetry interface is available from the real-time satellite ground system (as it is from ISI's EPOCH T&C ground system), TRECS can obtain a snapshot of the payload configuration directly from real-time, playback, or simulated telemetry. This configuration is loaded into TRECS for immediate use and also is saved to disk for future reference. To support contingency planning or planning a sequence of discrete payload reconfiguration activities, such as those that may occur in payload tests, TRECS can output the planned configuration as part of the command generation process previously described. Finally, as part of its comprehensive tool set, TRECS also compares two configurations and reports any differences encountered.

TRECS is fully compatible with the EPOCH T&C ground system. TRECS' highly configurable structure also can be adapted easily to support a wide variety of operating ground systems, if required. For maximum system flexibility, TRECS may be hosted on any UNIX or Windows (NT/2000) workstation and requires a JRE (Java Runtime Environment).

TMD (TRECS Model Definer)

The TMD provides a GUI (Graphical User Interface) to easily develop or modify highly structured XML (Extensible Markup Language) editor-based satellite models that can be loaded subsequently into TRECS for viewing and reconfiguring. TMD modifies the model by navigating through a series of buttons and pick list menus, intended to eliminate typographical errors, missing data errors, and other manual editing related errors. TMD also prevents incorrect option selection that ordinary manual text editing cannot.

TMD assumes a thorough understanding of the satellite being modeled and that the initial model is based on a valid payload configuration. The model must be assembled in an organized fashion and in a logical sequence. For example, the components (based on actual payload components) must be defined properly before they can be associated with a specific group or connected to another component. TMD's validation option can be selected prior to model generation to flag and present details on any problems or errors in the model. Thus, the model appears exactly as expected when ultimately displayed within the Mimic Editor window of the main TRECS application.

Because building the initial model is the most complex aspect of TRECS, TMD often functions as more of a pre-delivery, ISI-internal model development tool than as a combined TRECS delivery. In this case, the payload engineer enjoys the advantage of starting with an already assembled, valid model, rather than working from scratch.

Although TMD is used mainly for first-time, new model generation, TMD also edits existing model definitions and generates a modified new model that can be loaded into TRECS, as well. When a new model name is defined, the object types to incorporate into the model are assembled and selected. An existing model also can be copied in TMD, so that it can be edited or modified to include different data. The copy feature associates checkpoint features with the new model generation process, so that additional editing is not required.

While building a model, TMD can include a variety of standard satellite payload component types. TMD can include the following commandable switch type objects in a model: Coaxial C Switch, Waveguide C Switch, T Switch, Waveguide R Switch, Coaxial S Switch, and 2-3 Switch. The T Switch, the Coaxial C Switch, the Waveguide C Switch, and the Waveguide R Switch can be ganged to function together. All of these commandable switch components can have multiple configurations that can be selected in the Mimic Editor window of the TRECS GUI described in the next section.

TMD also can incorporate the following non-commandable objects in a model: Load, Hybrid, Simple Antenna, Receive-only Antenna, Transmit-only Antenna, Splitter, and Placeholder. The Simple Antenna object can be used simultaneously as both an input and an output in the model. The Receive-only Antenna has only an output mode. The Transmit-only Antenna has only an input mode. The Splitter and Placeholder objects are easily configurable and can be modified to include more ports than the default. Specifically, the Splitter can have more than 2 output ports, and the Placeholder can have more than 1 input and 1 output. The flexible Placeholder component can be used to model non-configuration related payload components, like receivers and amplifiers, thus, allowing end-to-end graphic representation of the entire payload.

The satellite telemetry and commands for each component also are specified in the model. The telemetry and command flat files from the ground system database are read by TRECS to extract all pertinent details required for command procedure generation.

Because one model in TRECS may contain multiple layers, the objects defined on each layer are organized into different Functional Groups. A Functional Group in TMD can be given a unique name and defined as a Payload, Channel Switching, Redundancy Ring, or Place Holder group. In the model, Functional Groups, Positions, and Settings are handled as Containers, which are considered large pieces of data that include other pieces of data. Containers can reside at any level in the hierarchy and include sets of unique commands and components. By selecting a sequence of buttons in TMD, such as Container, FunctionalGroups, Channel *name*, the mimic diagram layout can be assembled.

After inputs and outputs, components, and commands are specified, the model's connection paths can be defined. TMD presents a menu of component connections that can be coupled. As the components are connected working from left to right, TMD displays only the connections still available for use. This feature tracks the components used and ensures that the diagram will have a logical flow when brought up in TRECS. When all connections are completed, the model is generated and ready to be loaded into TRECS.

While there is an investment of time required to build an accurate model within the TMD environment; the return on this investment is significant. Several weeks of model definition are saved immediately. Additionally, each time that the model is used with TRECS, the return on that initial investment grows; because using a complete, accurate model makes it easy to quickly load, reconfigure, and generate the associated commands or procedure for the lifetime of the satellite.

TRECS

The TRECS Main GUI (see Figure 1) enables payload engineers to quickly and easily view and plan a payload reconfiguration in a user-friendly, graphical environment, based on the models created by the TMD. Configuration files, either saved by the user or derived automatically from telemetry, can be loaded to reflect the desired payload state and, thus, offer a convenient baseline from which to begin work. When the configuration data has been loaded, the user can either use the system to visualize the various communication paths, or plan a payload reconfiguration with a few mouse clicks. TRECS' powerful AutoSolve feature helps users to wade through the sometimes near limitless change options by narrowing those options to changes meeting desired constraints. Finally, it can output a procedure or command list to implement the desired reconfiguration in a format ready for execution on the real-time ground system.

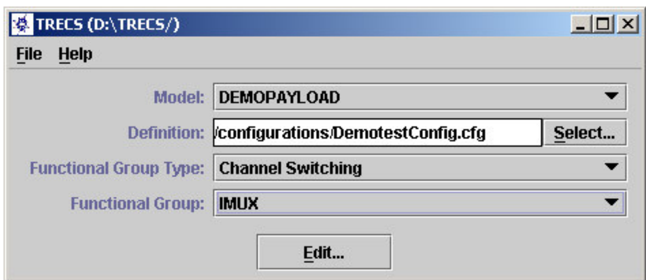

Figure 1. The TRECS Main GUI
The TRECS GUI enables users to select the model and associated parameters.

TRECS Configuration and Snapshot Capability

When selecting a model from the main menu, the user chooses a configuration definition to give TRECS a starting point for the model diagram to be viewed and modified: a "configuration," or a "snapshot" file. Many different configuration and snapshot files may be saved for one model. Each file may have a different status of the model's components.

A configuration file is a previously saved payload configuration that can be retrieved at a future time. Configuration files can be stored and saved conveniently at any point in the process of working with TRECS. When several iterations or phases are needed before implementing a final payload configuration, the user can organize their work cleanly into different configuration files and can quickly access the desired states of the components in the model without uncertainty or difficulty. This ability is especially valuable when preparing for critical operations situations, such as payload shed on a spacecraft.

Snapshot files include the actual (at the time of the snapshot) states of components captured from live or simulated telemetry data from the ground system. TRECS can communicate with the real-time ground system and capture the current settings of each of the model's components. A library of configurations can be created during simulation or test phases that can be used for training, anomaly investigation, and recovery procedure development. These snapshot configurations guarantee that the engineer will have the accurate initial values required before proceeding to make the desired configuration changes.

If there is any uncertainty about which configuration file or snapshot file should be loaded in TRECS, the user can compare two existing files quickly. With this option, two files with different names are checked for discrepancies. Any differences found will be flagged and presented in a window. The Compare option always can be used immediately after a new snapshot file is created to verify that any configuration file assumed to be the current configuration truly represents the spacecraft state. This feature ensures that the correct mimic diagram is loaded in TRECS before making any edits or changes.

The Functional Group Type and corresponding Functional Group Name also must be selected before the model is viewed. As mentioned previously for TMD, a model may contain several different Functional Groups with unique names. Each Functional Group listed will represent a different spacecraft payload model or subsystem model. These selections in TRECS allow the user to go directly to the spacecraft payload model, or section of that model that requires attention, rather than navigating through multiple layers to reach the desired portion of the model.

TRECS Mimic Editing

The TRECS MIMIC Editor window pops up a visual representation matching the defined satellite payload schematic for the specified functional group. Initially, all input and output paths are indicated in bold, black lines. The input and output paths along the left and right sides of the diagram can be selected by clicking once with the mouse. The path that has been selected will change from black to a bright color. Multiple paths can be selected. A new color will be assigned for each path, so that the user may distinguish clearly the flow of each separate path on the diagram.

The Mimic Editor displays the diagram from the chosen model, which may be scaled to fit entirely within the window. It includes an option to display a legend at the bottom of the window that prominently indicates all payload component types applicable to the displayed group. The commandable objects appear in a bold black outline, while the non-commandable objects are displayed in a light gray outline, indicating that they do not have an alternate setting. Optionally, the user also may show the component names and pin numbers on the Mimic Editor display.

TRECS offers a choice of two main methods to edit the mimic diagrams: individual component editing, or automatically solving for component changes, given specified input and output paths.

Method 1 – Individual Component Editing

On the diagram, all components with a bold black outline can be changed individually to an alternate configuration by clicking once with the mouse directly on top of the component to be edited. If a component has more than 2 different settings, each sequential mouse click on the component corresponds to the next setting. When a component is clicked to a new configuration, its color changes from black to magenta, which is an immediate flag to the user that this object has been edited from its original configuration on the diagram. If any unintentional or inadvertent changes are made in the editing process, the unedited mimic diagram state may be restored.

Any paths in the functional group affected by changing the edited component automatically change to reflect the new state. Figure 2 illustrates a sample channel-switching diagram with a changed configuration in the TRECS Mimic Editor. In this case, the input indicated by A2 was switched to the output indicated by Z1, so the A1 and Z2 input routing changed automatically.

Any commandable component on the diagram that is displayed with a gray outline indicates that it has been flagged as failed or reserved in the model. Therefore, these components are protected in TRECS against any change. If the user clicks on these components, nothing will change on the diagram.

If an object appears on the diagram with a bright blue outline and a functional group name above it, the user can click on this object to drill down into that sub-layer of the payload or channel-switching diagram. When any edits are made to one layer, subsequent layers affected by the modification will reflect automatically the new path changes. The Zoom Out feature allows the user to navigate easily back to the previous level to view the changes. TRECS will accommodate any number of levels of nested functional groups defined in the model.

Figure 3 shows a TRECS payload diagram top layer that includes functional groups with sub-layers. The long vertical rectangular objects on the diagram are examples of functional groups defined in this model as components. Clicking on these objects accesses the component diagrams in the Receivers, IMUX (Input Multiplexer), TWTAs, or OMUX (Output Multiplexer) functional groups. In this particular example, note that the IMUX functional group object has 3 input ports and 6 output ports with specific paths defined. Clicking on this object maintains and displays these paths in the new window with the IMUX mimic diagram and its switch components. Figure 4 shows the IMUX functional group. While displaying the detailed contents of the functional group, any switch on this layer can be edited so that the alternate configuration is chosen. The paths change corresponding to the configuration chosen. When the user zooms back out of that layer, the original diagram from Figure 3 is displayed again, along with any changes made to the paths in the functional group.

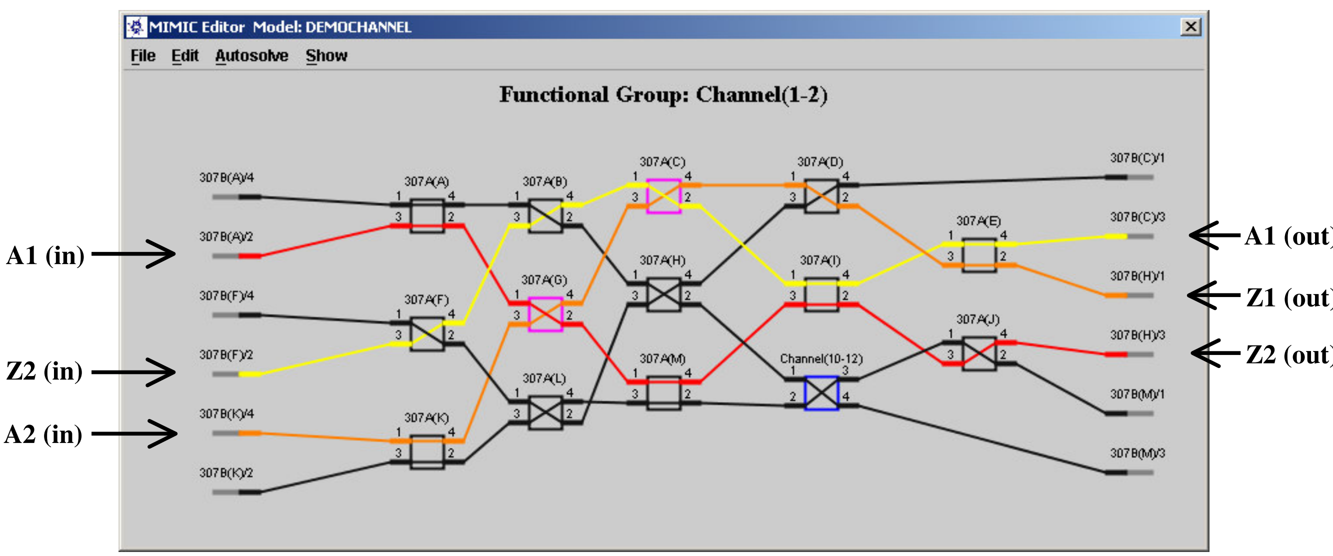

Figure 2. Sample Channel Switching Diagram in the TRECS Mimic Editor

If components are edited directly in the TRECS Mimic Editor, their associated input and output paths will change automatically on the diagram.

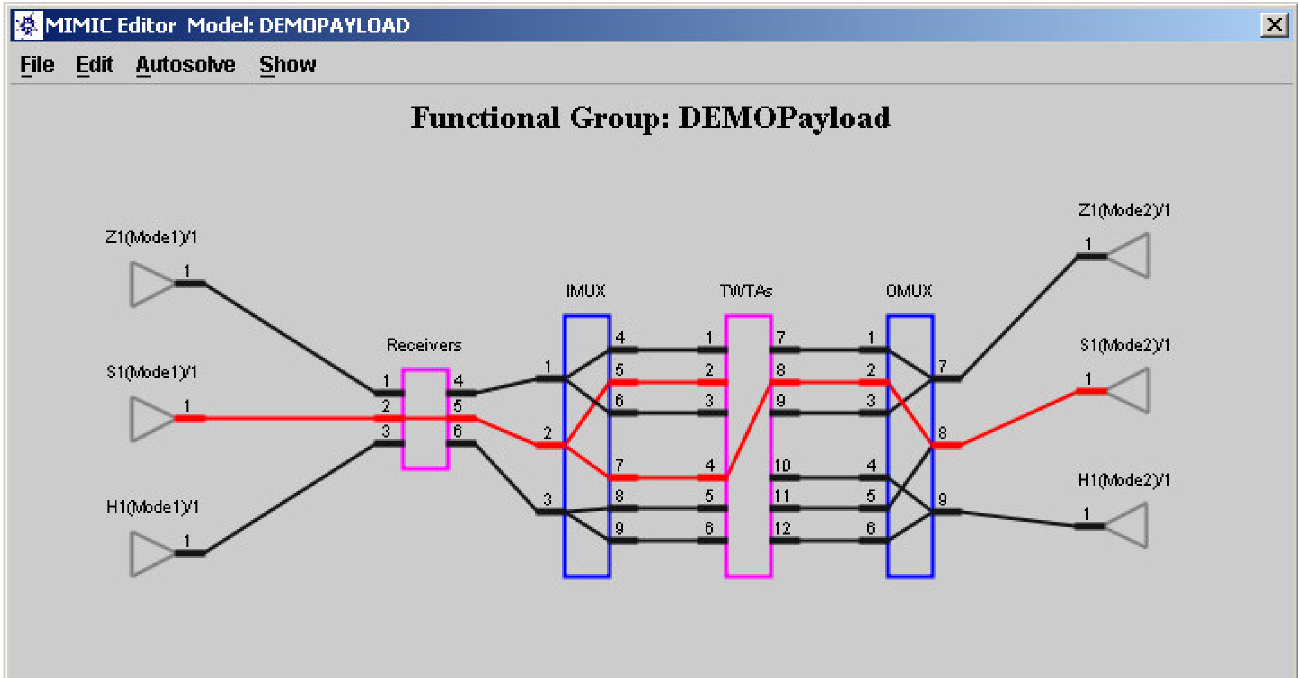

Figure 3. Channel-Switching Layer of a TRECS Mimic Diagram
Objects on a mimic diagram can be components or other functional groups. Functional groups appear with a bright blue outline.

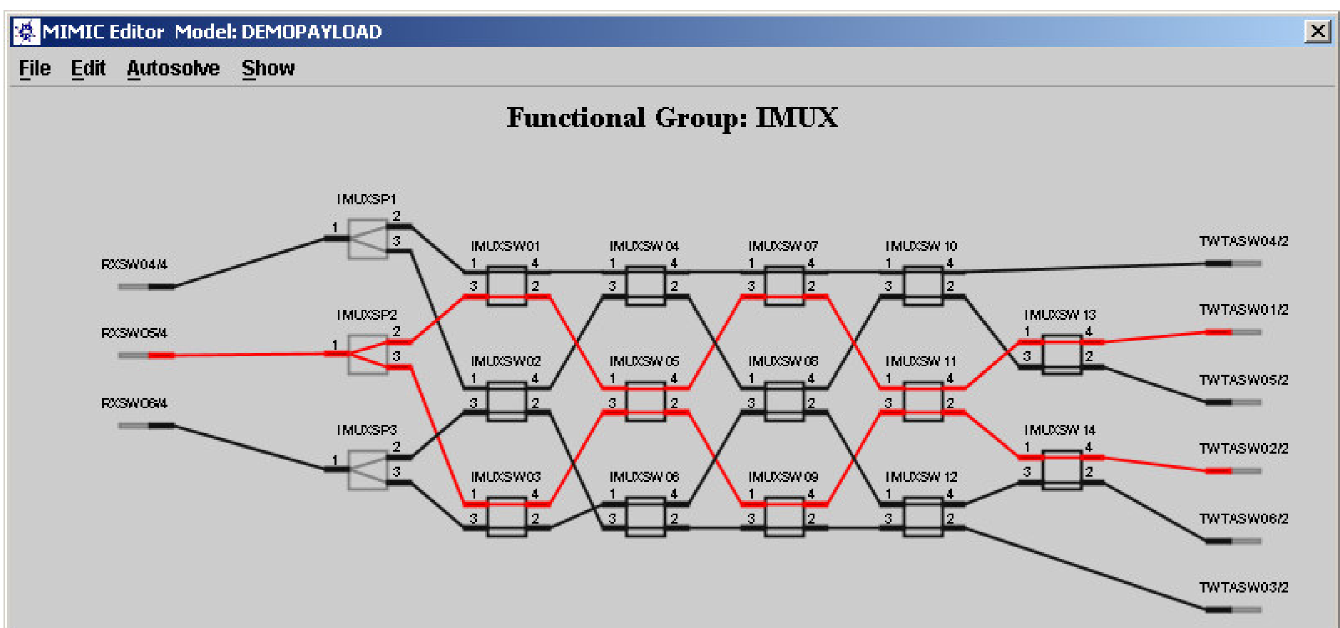

Figure 4. Sub-Layer IMUX of the TRECS Mimic Diagram in Figure 3
When a functional group object on a top layer is clicked on a mimic diagram, the components contained within that layer are displayed and can be edited. This is the IMUX functional group from Figure 3.

Method 2 – Automatically Solving for Component Changes

For cases where the user must define specific inputs and outputs rather than editing individual components on the diagram, TRECS employs an AutoSolve feature to find the best solution, given a set of constraints. The currently available algorithm, called FindPaths, is operational for the Channel Switching, Redundancy Ring, and Place Holder groups within the payload. For a set of paired inputs and outputs defined by the user, this algorithm finds all possible combinations of changes accommodating that request. The affected components in the functional group are changed automatically in the resulting solution(s).

The AutoSolve window permits the user to pair inputs and outputs according to the desired configuration by selecting from a pick list menu. The menu updates and removes the items automatically after they have been chosen. Optionally, the user may specify all paths or only some paths. If more paths are specified, fewer solutions will be returned, as any selected path becomes a requirement for the solutions.

AutoSolve constraint options include Minimize Changes, Limit Changes to *n*, Search Embedded Groups, and Take First *n* Solutions. Any one or more of these constraints can be selected for AutoSolve to search for the best solutions. If required, constraints can be added to the system relatively easily to meet specific program requirements.

Upon completion of the AutoSolve process, the solutions meeting the specified parameters are presented on the screen. Each solution contains both the changes to the group's top-level components and the nested solutions to any embedded functional groups. They can be examined individually and accepted or rejected from a selection menu on the window.

The chosen solution, either from AutoSolve or from the component editing method, is ready to be used on the T&C ground system with one quick step. The Generate Commands option from the Mimic Editor window will translate the changes made within TRECS into a valid output format with all the commands and operations required for the new changes to take effect. The user may generate a command list format, an operational procedure format with both commands and associated automated telemetry checking, or both formats in two separate outputs.

In addition, the user can elect to append the current configuration changes to an existing procedure or command list. This feature allows the user to develop very complex payload procedures in a modular, controlled process.

Also, for convenience within the TRECS Mimic Editor window, a command pick list enables the user to append additional valid satellite commands to the end of the command list or procedure. This feature provides the ability to incorporate other associated commanding, such as transponder gain and level setting, unit on/off commands, etc., directly from the TRECS tool to provide the complete procedures. When combined with the append capability, it enables the TRECS tool to construct large, complex reconfiguration procedures.

TRECS Output

The resulting payload changes that are planned in TRECS can be saved for later reference by TRECS or output in a procedure file or command list for subsequent execution via the real-time ground system. By using this feature, operators not only minimize the time and effort required to build procedures to perform this task manually, but in the case of an anomaly, may further reduce the duration of time that a customer experiences a communications outage.

TRECS currently can generate a procedure compatible with ISI's EPOCH 2000 ARES (Automated Real-Time Execution Suite) procedure development and execution platform. However, TRECS' modular design enables it to be adapted easily to support other required procedure or commanding tools.

TRECS Procedure Output for ARES

The procedure output file for ARES contains all commands and telemetry checks (i.e., current real-time telemetry value compared against expected value in the procedure) required to make the changes from TRECS take effect. ARES includes both the ARES Shell for procedure execution and the ARES Procedure Builder for creating and editing procedures. The ARES Shell provides a visual, flexible, and easy-to-use automated procedure execution capability. It has been designed so that the procedure developer and the satellite operator require no special programming-related training to develop and run the spacecraft operations procedures.

ARES is developed using the Tcl/Tk (Tool Control Language/Toolkit) open source scripting language used throughout the industry in a variety of applications, thus, providing a stable, proven platform. The ARES Shell interfaces to EPOCH T&C via EPOCH T&C's Stream Services Interface [EPOCH T&C's open API (Application Programming Interface) used by both ISI and customers for interfacing applications to the EPOCH T&C system].

The ARES GUI format for monitoring procedure execution is modeled after typical spacecraft flight operations procedures. An example is shown in Figure 5. The standard configuration includes an outline of the procedure and a detailed view of the paper procedure, both of which are highlighted showing the procedure execution status.

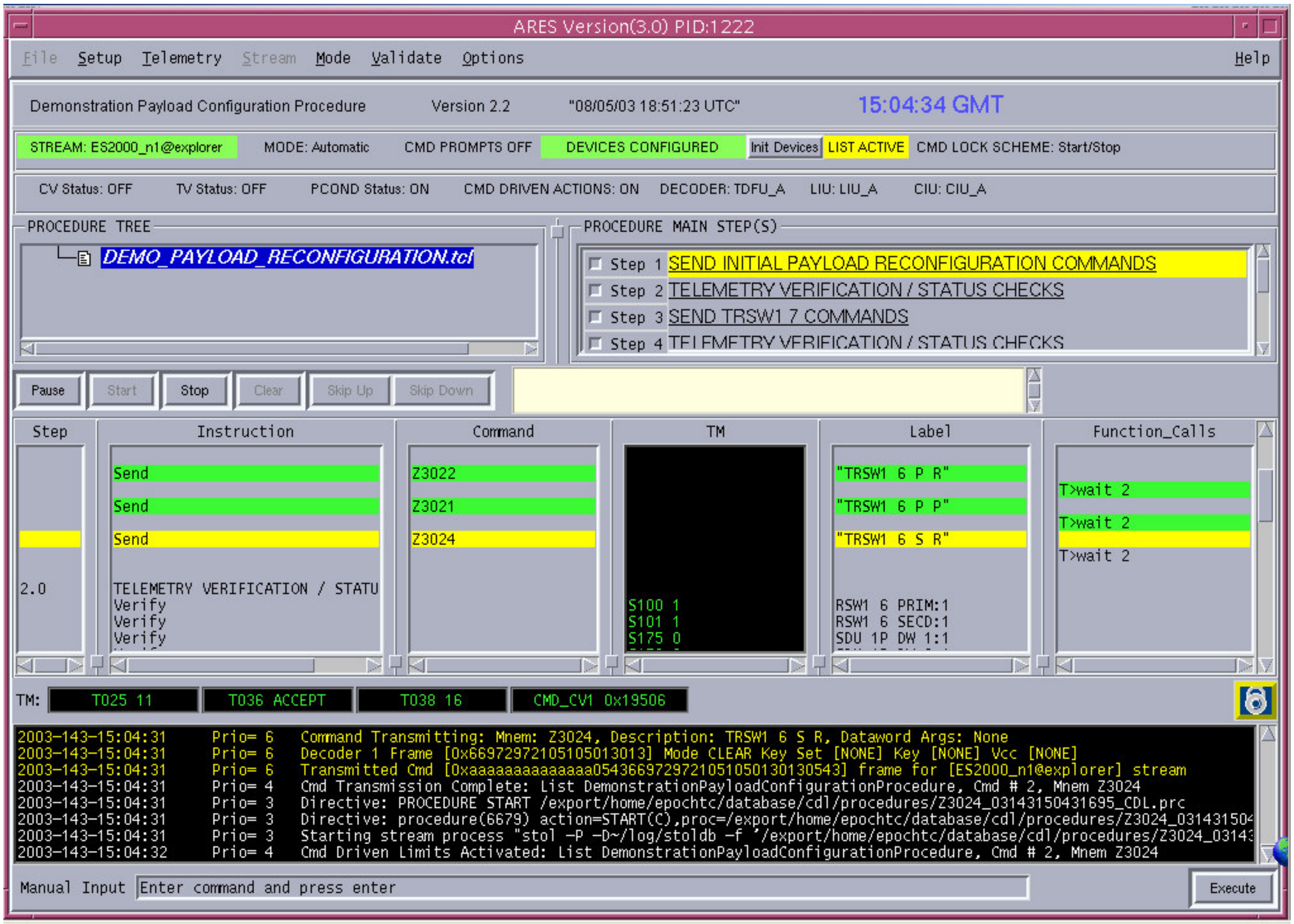

Figure 5. The ARES Main GUI

ARES, designed for typical spacecraft operations procedures, readily executes procedure output files generated by TRECS.

The procedure format is intended to minimize operator training and to maximize their comfort with the system, as well as to reduce the possibility for operator error. The clear-cut layout of these procedures allows users to reference quickly all required real-time telemetry point values, as well as required command data words, which are presented clearly within the procedure display.

While using the ARES Shell, the user has easy access to all ground system indicators and settings required for procedure execution. All spacecraft event messages regarding telemetry and commands, as well as the procedure execution status in general, appear in the event display area toward the bottom of the window. The current values of several key spacecraft telemetry parameters also are present constantly for health and safety monitoring throughout the procedure. Any spacecraft-specific set-up devices and commanding flags for command verification; telemetry verification, or pre-requisite checks can be selected from the main menu. Any system directives that must be executed outside of the procedure's scope may be entered by the user in the input field at the bottom of the window and quickly transmitted without having to bring up any other windows or applications, and without exiting the procedure. Because all ingredients needed to execute and monitor the procedure effectively are available immediately on the ARES Shell, it is unnecessary to bring up multiple resources during procedure operation.

If the detailed view of the procedure that the ARES Shell displays is not required in operations, ARES procedures can also be started via a command line to support automatic procedure starting and execution. For example, when the procedure is generated from TRECS, ARES could potentially be called from a scheduling tool with the procedure name and all associated options and arguments. If ARES is invoked without a GUI, only a small status window appears on the workstation to represent the ARES procedure execution. This offers a means to access the procedure in the event of a problem or any necessary user intervention.

The ARES procedure file is a tab-delimited text file corresponding to the columns in the detailed display area of the ARES Shell. The step numbers, detailed instructions, commands, telemetry values, and telemetry checks are all ordered in the correct sequence of execution corresponding to the changes requested by the user in TRECS. Although no changes are required for execution after generating the procedure in TRECS, the files can be edited easily using the ARES Procedure Builder. This editing tool has a convenient spreadsheet format that parallels the ARES Shell layout to ensure accurate changes and to check the validity of all spacecraft data in the procedure.

Even after an ARES procedure has been generated in TRECS, additional telemetry checks and commands can be inserted into the procedure using the ARES Procedure Builder. The pre-defined pick list menus for spacecraft telemetry and commands contain all applicable selections and fields needed to update the procedure safely and precisely. To further protect against errors while creating or editing the procedure, all of the inserted telemetry checks and commands can be validated against the spacecraft database files used by the real-time system. Any discrepancies are flagged and presented in a window as a warning of the failures.

The ARES Procedure Builder also allows the user to incorporate function calls for more flexibility and logic for flow control in the procedure. A comprehensive function library available when creating and editing ARES procedures includes, IF-THEN-ELSE structures, variable assignments, relative and absolute waits, command argument file loads, stop, pause, user prompts, and sub-procedure calls, among other functions. The sub-procedure function can be used to break up a complicated, single level procedure generated by TRECS into multiple, streamlined, nested procedures, which can be considered more concise and clear in operations.

Although ARES is fully compatible with TRECS for payload reconfiguration operations, it also has been used for many other types of complex spacecraft operations procedures, such as stationkeeping, Earth sensor inhibitions, battery reconditioning, etc. Its many configurable and adaptable options make ARES desirable as a procedure execution platform, so it is not tied exclusively to execute TRECS output procedures. However, the combined effectiveness of TRECS and ARES provides complete, end-to-end spacecraft payload reconfiguration planning and execution features with minimal expenditure of labor, time, and resources.

TRECS' overall ability to reconfigure satellite payloads clearly and quickly, while maintaining optimum precision, now enables satellite operators to minimize the impact of any unexpected communications outages to their customers. Because TRECS is model driven, satellite operators can do this for multiple satellites in their own environment, without requiring outside custom software modifications. Additionally, the increased efficiency gained by using TRECS greatly reduces the possibility of errors during reconfiguration, decreasing outages on the whole.

Information

This paper briefly summarizes the abilities of TRECS and its associated outputs. For detailed information on the TRECS, EPOCH, or ARES products mentioned in this paper, please, contact:

Integral Systems, Inc.
5000 Philadelphia Way, Suite A
Lanham, Maryland 20706-4417 (USA)
Tel: (301) 731-4233
Fax: (301) 731-9606
Email: sales@integ.com
Web: http://www.integ.com

The following are trademarks or registered trademarks:

TRECS—Integral Systems, Inc.
ARES—Integral Systems, Inc.
EPOCH T&C—Integral Systems, Inc.
UNIX—UNIX Systems Laboratories, Inc.